

IN THE CLAIMS:

Please amend claims 1, 4-5, 8, 11-12, 15-16, 20-22, and 25, and cancel claims 3, 10, and 24 as follows:

1. (Currently Amended) An input/output hub, comprising:
 - an inbound ordering queue (IOQ) to receive inbound transactions, wherein all read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;
 - an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;
 - an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol, wherein the unordered protocol is a coherent interface, and completions of the inbound transactions, and to issue a write completion for a posted write;
 - an OOQ read bypass buffer to receive read transactions pushed from the OOQ to permit the posted writes and the read/write completions to progress through the OOQ; and
 - an unordered domain to receive the inbound transactions transmitted from the IOQ and to receive the outbound transactions transmitted from ~~an~~ the unordered protocol.

2. (Original) The input/output hub according to claim 1, wherein the IOQ does not permit the inbound read and write transactions to bypass inbound write data.

3. (Canceled)

4. (Currently Amended) ~~The input/output hub according to claim 3, wherein the coherent interface is a Scalability Port.~~

An input/output hub, comprising:

an inbound ordering queue (IOQ) to receive inbound transactions, wherein all read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol, wherein the unordered protocol is a Scalability Port, and completions of the inbound transactions, and to issue a write completion for a posted write;

an OOO read bypass buffer to receive read transactions pushed from the OOO to permit the posted writes and the read/write completions to progress through the OOO; and

an unordered domain to receive the inbound transactions transmitted from the IOQ and to receive the outbound transactions transmitted from the unordered protocol.

5. (Currently Amended) An input/output hub, comprising:

an ordered domain, including:

an inbound ordering queue (IOQ) to receive and transmit inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the ~~destination~~, destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the ~~IOQ~~, IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol and completions of the inbound transactions, and to issue a write completion for a posted ~~write~~, and write;

an OOQ read bypass buffer to receive read transactions pushed from the OOQ to permit the posted writes and the read/write completions to progress through the OOQ; and

an unordered domain, in communication with an unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered protocol, and

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain, wherein the unordered protocol is a coherent interface.

6. (Original) The input/output hub according to claim 5, further including at least one Producer-Consumer ordered interface in communication with the ordered domain.

7. (Original) The input/output hub according to claim 6, further including an input/output device connected with the Producer-Consumer ordered interface.

8. (Currently Amended) The input/output hub according to claim 7, ~~further including~~ coupled to an intermediary device that is external to the input/output hub, the intermediary device interconnecting the Producer-Consumer ordered interface and an input/output device.

9. (Original) The input/output hub according to claim 7, wherein the input/output device is a Peripheral Component Interconnect (PCI) device.

10. (Canceled)

11. (Currently Amended) ~~The input/output hub according to claim 10, wherein the coherent interface is a Scalability Port.~~

An input/output hub, comprising:

an ordered domain, including:

an inbound ordering queue (IOQ) to receive and transmit inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol and completions of the inbound transactions, and to issue a write completion for a posted write;

an OOQ read bypass buffer to receive read transactions pushed from the OOQ to permit the posted writes and the read/write completions to progress through the OOQ; and

an unordered domain, in communication with an unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered protocol, and

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain, wherein the unordered protocol is a Scalability Port.

12. (Currently Amended) An input/output system, comprising:

an ordered domain, including:

an inbound ordering queue (IOQ) to receive and transmit inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does

not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the ~~destination~~, destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the ~~IOQ~~, IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol, wherein the unordered protocol is a coherent interface, and completions of the inbound transactions, and to issue a write completion for a posted ~~write~~, write;

an OOO read bypass buffer to receive read transactions pushed from the OOO to permit the posted writes and the read/write completions to progress through the OOO; and

an unordered domain, in communication with ~~[[an]]~~ the unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered ~~protocol~~, and protocol;

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain;

a Producer-Consumer ordered interface in communication with the ordered domain; and

an input/output device connected with the Producer-Consumer ordered interface; and

~~a coherent interface within the unordered protocol in communication with the unordered domain.~~

13. (Original) ~~The input/output system according to claim 12, wherein the coherent interface is a Scalability Port.~~

An input/output system, comprising:

an ordered domain, including:

an inbound ordering queue (IOQ) to receive and transmit inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol, wherein the unordered protocol is a Scalability Port, and completions of the inbound transactions, and to issue a write completion for a posted write;

an OOO read bypass buffer to receive read transactions pushed from the OOO to permit the posted writes and the read/write completions to progress through the OOO; and

an unordered domain, in communication with the unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered protocol;

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain;

a Producer-Consumer ordered interface in communication with the ordered domain; and

an input/output device connected with the Producer-Consumer ordered interface.

14. (Original) The input/output system according to claim 12, wherein the input/output device is a Peripheral Component Interconnect (PCI) device.

15. (Currently Amended) The input/output system according to claim 12, ~~further including~~ coupled to an intermediary device external to the input/output system, the intermediary device interconnecting the Producer-Consumer ordered interface and the input/output device.

16. (Currently Amended) An input/output system, comprising:
an ordered domain having a first functional block and a second functional block, wherein the first functional block and the second functional block each include:

an inbound ordering queue (IOQ) to receive inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit

subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the ~~destination~~, destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the ~~IOQ~~, IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol, wherein the unordered protocol is a coherent interface, and completions of the inbound transactions, and to issue a write completion for a posted ~~write~~, write;

an OOO read bypass buffer to receive read transactions pushed from the OOO to permit the posted writes and the read/write completions to progress through the OOO; and

an unordered domain, in communication with ~~[[an]]~~ the unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered ~~protocol~~, and protocol;

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain;

a first Producer-Consumer ordered interface in communication with the first functional block;

a first input/output device connected with the first Producer-Consumer ordered interface;

a second Producer-Consumer ordered interface in communication with the second functional block; and

a second input/output device connected with the second Producer-Consumer ordered interface; and

~~a coherent interface within the unordered protocol in communication with the unordered domain.~~

17. (Original) ~~The input/output system according to claim 16, wherein the coherent interface is a Scalability Port~~

An input/output system, comprising:

an ordered domain having a first functional block and a second functional block, wherein the first functional block and the second functional block each include:

an inbound ordering queue (IOQ) to receive inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from an unordered protocol, wherein the unordered protocol is a Scalability Port, and

completions of the inbound transactions, and to issue a write completion for a posted write;

an OoQ read bypass buffer to receive read transactions pushed from the OoQ to permit the posted writes and the read/write completions to progress through the OoQ; and

an unordered domain, in communication with the unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered protocol;

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain;

a first Producer-Consumer ordered interface in communication with the first functional block;

a first input/output device connected with the first Producer-Consumer ordered interface;

a second Producer-Consumer ordered interface in communication with the second functional block; and

a second input/output device connected with the second Producer-Consumer ordered interface.

18. (Original) The input/output system according to claim 16, wherein the first input/output device is a Peripheral Component Interconnect (PCI) device.

19. (Original) The input/output system according to claim 16, wherein the second input/output device is a Peripheral Component Interconnect (PCI) device.

20. (Currently Amended) The input/output system according to claim 16, ~~further including~~ coupled to a first intermediary device external to the input/output system, the intermediary device interconnecting the first Producer-Consumer ordered interface and the first input/output device.

21. (Currently Amended) The input/output system according to claim 16, ~~further including~~ coupled to a second intermediary device external to the input/output system interconnecting the second Producer-Consumer ordered interface and the second input/output device.

22. (Currently Amended) A computer system, comprising:

- a plurality of processor units having access to caches;
- a main memory;
- a coherent interface to maintain coherency between the processor units and their caches;
- a scalability node controller interconnecting the processor units, the main memory, and the coherent interface to control interface therebetween; and
- an input/output hub in communication with the coherent interface, including:
 - an inbound ordering queue (IOQ) to receive inbound transactions,

wherein all read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from the coherent interface and completions of the inbound transactions, and to issue a write completion for a posted write;

an OOQ read bypass buffer to receive read transactions pushed from the OOQ to permit the posted writes and the read/write completions to progress through the OOQ; and

an unordered domain to receive the inbound transactions transmitted from the IOQ and to receive the outbound transactions from the coherent interface.

23. (Original) The computer system according to claim 22, wherein the IOQ does not permit the inbound read and write transactions to bypass inbound write data.

24. (Canceled)

25. (Currently Amended) ~~The computer system according to claim 22, wherein the coherent interface is a Scalability Port.~~

A computer system, comprising:

a plurality of processor units having access to caches;

a main memory;

a Scalability Port to maintain coherency between the processor units and their caches;

a scalability node controller interconnecting the processor units, the main memory, and the Scalability Port to control interface therebetween; and

an input/output hub in communication with the Scalability Port, including:

an inbound ordering queue (IOQ) to receive inbound transactions,

wherein all read and write transactions have a transaction completion, peer-to-peer

transactions are not permitted to reach a destination until after all prior writes in the IOQ

have been completed, and a write in a peer-to-peer transaction does not permit

subsequent accesses to proceed until the write is guaranteed to be in an ordered

domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the

IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions from

the Scalability Port and completions of the inbound transactions, and to issue a write

completion for a posted write;

an OOO read bypass buffer to receive read transactions pushed from the

OOQ to permit the posted writes and the read/write completions to progress through the

OOQ; and

an unordered domain to receive the inbound transactions transmitted from the

IOQ and to receive the outbound transactions from the Scalability Port.

///

///

///

///

///

///